REMARKS

Applicant appreciates the Examiner's reconsideration of the present application. The present application contains six independent claims 1, 9, 35, 36, 43 and 44.

The Invention

The present invention is directed to the transformation of model objects between layers in a metadata model (e.g., claims 1 and 9) or within a layer of the metadata model (e.g., claim 9)). As recited in claim 1, a model object in a layer with a lower degree of abstraction is transformed and a new model object is created in a higher layer with a higher degree of abstraction. Thus, an object in a source database is represented as multiple model objects in the metadata model.

Rejection under 35 USC 103(a) to claims 1-8, 35 and 43-44

The Examiner has rejected claims 1-8, 35 and 43-44 under 35 U.S.C. 103(a), stating that these claims are unpatentable over Mullins (US Patent No. 5,857,197) in view of Anderson et al (US Patent No. 5,799,310 - hereinafter called "Anderson"). Applicant respectfully requests reconsideration of the rejection based on the reasons set out below.

The present invention as recited in independent claims 1, 35, 43 and 44 is for transforming a metadata model. The present invention as recited in independent claims 1, 35, 43 and 44 transforms a metadata model by obtaining information from objects in a layer of the model, abstracting that information by adding business intelligence, and creating objects in the same or higher layer of that model. The present invention as recited also transforms a metadata model by obtaining information from objects in a layer of the model, modifying the obtained information, and transforming the objects based on the modified information.

Mullins does not disclose any metadata model having multiple layers containing model objects of different degrees of abstraction.

PAGE 15/23 * RCVD AT 9/28/2004 4:02:29 PM [Eastern Daylight Time] * SVR:USPTO-EFXRF-1/14 * DNIS:8729306 * CSID:7709840098 * DURATION (mm-ss):06-22

Mullins discloses a 3-tier architecture for accessing data stored in a data store over a distributed network (column 3, line 63). A 3-tier architecture is commonly used to allow a client, such as a user browser, to access a database through a server having a business logic. In a 3-tier architecture, the user browser (e.g., object application 101 in Fig. 1 of Mullins) sends a request for data to an application having a business logic (e.g., adapter abstraction layer 600 in Fig. 1 of Mullins), which modifies the request and sends the modified request to a database (data stores 302, 312 in Fig. 1 of Mullins) to retrieve the requested data. The retrieved data is sent back to the application, which in turn modifies and sends the data to the user browser.

The 3-tier architecture is not a metadata model, but rather it is a service architecture of different hardware components. In Figure 1 of Mullins, there seems to be three "layers" in the 3-tier architecture. However, these "layers" do not correspond to layers in a metadata model, because each "layer" in the 3-tier architecture represents a hardware component in the architecture.

Mullins discloses the user of meta data 201. However, Mullins describes the meta data 201 only as "data used to describe other data" (column 4, line 34). Mullins does not disclose any detail of meta data 201 or any meta data model. No layers of different degrees of abstraction are shown in the meta data 201.

Accordingly, Mullins does not disclose any metadata model having multiple layers containing model objects of different degrees of abstraction.

Furthermore, Mullins does not disclose any metadata model transformer, or transformation of objects in a metadata model.

The Examiner has suggested that Mullins discloses in column 4, lines 33-67 transformation of a model object in a lower layer of a metadata model and creation of a model object in a higher layer of the metadata model. It is respectfully submitted that this section of Mullins discloses transformation of data, rather than transformation of model objects in a metadata model.

Mullins' system uses an object schema 200 including meta data 201 (column 4, lines 1-2). The object application 101 accesses the object schema

PAGE 16/23 * RCVD AT 9/28/2004 4:02:29 PM [Eastern Daylight Time] * SVR:USPTO-EFXRF-1/14 * DNIS:8729306 * CSID:7709840098 * DURATION (mm-ss):06-22

200 through the abstract layer 600 "which does conversion between object and non-object views" (column 4, lines 9-10). For the conversion, Mullins uses "meta data 201 to define how to access and convert non-object data store content 304 to objects and back" (column 4, lines 32-36). This is conversion of non-object data to object data in order that the object application can understand the data. This data conversion does not correspond to the transformation of model objects in a metadata model, as recited in the claims of the present application.

As the Examiner has kindly indicated, Mullins does not teach means for abstracting the information by adding business rules for representing a business concept.

The Examiner has attempted to overcome this deficiency of Mullins by combining the teaching of Anderson et al (US Patent No. 5,799,310 - hereinafter called "Anderson") with Mullins.

Anderson proposes to store attributes of an object, such as video or audio, in multiple tables, and provide relational extenders in the first table to point the relevant entries in the other tables storing the relevant attributes.

The Examiner uses column 7, lines 47-50 in Anderson which reads "the user's application data table and object data are the bottom layer of the system. Metadata tables are maintained to manage and access the data tables". The "user's application data table" described is a business table 312 shown in Figures 4 and 6. The "metadata tables" described is the Base Metadata Table 412 and the Attribute Metadata Table 418 shown in Figure 4.

The business table 312 has multiple columns, one of which is used to store object handles 310 to point to entries in the Base Metadata Table 412 and the Attribute Metadata Table 418 (column 6, lines 29-38). The Base Metadata Table 412 is used to store attributes that are common to all objects, regardless of the data type, and the Attribute Metadata Table 418 is used to store attributes that are specific to the object data type (column 6, lines 45-62). The attributes that are common to all objects include the name of the person creating the object, the date of creation, and the location of the actual object data (column 6, lines 46-50). The attributes that are specific to the object data type include the

PAGE 17/23 * RCVD AT 9/28/2004 4:02:29 PM [Eastern Daylight Time] * SVR:USPTO-EFXRF-1/14 * DNIS:8729306 * CSID:7709840098 * DURATION (mm-ss):06-22

number of frames and the frame size for a video data type, and the duration of play for an audio data type (column 6, lines 59-62). That is, the metadata tables 412, 418 disclosed by Anderson are extension of the business table 312 in order to store part of data type values (column 6, lines 45-46).

While Anderson indicates the business table being in "the bottom layer", this statement does not represent that the business table has a lower abstraction and the metadata tables have higher abstractions, or the "layer" contain model objects. Figure 5 is a schematic view to show the data flow in different service "layers". This schematic view does not correspond to a metadata model as recited in the claims of the present application.

Accordingly, Anderson does not disclose any metadata model having multiple layers containing model objects of different degrees of abstraction.

Furthermore, Mullins does not disclose any metadata model transformer, or transformation of objects in a metadata model.

The Examiner has suggested that column 6, lines 39-44 of Anderson discloses addition of the business rules in order to manipulate the object data. In this section, Anderson discloses that a business table 312 column containing the object handler 310 is defined as the corresponding complex data type or UDT (user defined type), e.g., audio or video, and that the interface uses the object handle as a parameter to manipulate the object data. This is not modeling of metadata, or adding business rules to an object in a metadata mode.

As discussed above, the object handle 310 points its relevant entry in the metadata tables. Thus, the interface can access and manipulate the object data in the Mullins' system. This section simply describes the use of the object handler 310, i.e., a pointer, to points an entry in the Attribute Table 314 (column 6, lines 34-35). For example, when the user requests the name of the person creating an object represented by a selected row in the business table 312, the interface uses the object handle 310 stored in the last column of the selected row to access the Attribute Table 314 which stores the requested name (column 6, lines 45-50).

PAGE 18/23 * RCVD AT 9/28/2004 4:02:29 PM [Eastern Daylight Time] * SVR:USPTO-EFXRF-1/14 * DNIS:8729306 * CSID:7709840098 * DURATION (mm-ss):06-22

The use of the object handle 310 allows manipulating the object data, but it does not add any business rules to model objects to abstract the information of the model object, as recited in the claims of the present application.

Both Mullin and Anderson disclose the use of metadata. However, neither Mullin nor Anderson discloses or suggests provision of a metadata model having layers of different degrees of abstraction, or transformation of such a metadata model. Accordingly, even if one skill in the art attempts to combine Mullin with Anderson, he would still fail to obtain a metadata model transformer as recited in the claims of the present application.

Regarding to dependent claims 2-8, the Examiner has suggested, using the paragraphs in column 4, lines 33-67, that Mullins teaches the transformations recited in these claims. As discussed above, in column 4, lines 33-67, Mullins discloses the conversion between object data and non-object data, which is different from transformation of model objects in a metadata model. Mullins does not teach any metadata model having model objects in higher and lower layers, and thus Mullins does not disclose or suggest any means for obtaining information from a model object in the lower layer, means for modifying the obtained information or means for transforming the model object in the lower layer, as required in claim 2.

Similarly, Mullins does not disclose or suggest any means for obtaining information from a model object or means for modifying a model object in any layer in a metadata model, as required in claims 3-8.

Claims 43 and 44 correspond to claim 1, and thus the same arguments discussed above apply to these claims.

Therefore, Applicant trusts that claims 1-8 and 43 and 44 have patentably distinguished over Mullins and Anderson.

PAGE 19/23 * RCVD AT 9/28/2004 4:02:29 PM [Eastern Daylight Time] * SVR:USPTO-EFXRF-1/14 * DNIS:8729306 * CSID:7709840098 * DURATION (mm-ss):06-22

Rejection under 35 USC 103(a) to claims 9-21, 24-33 and 36-42

The Examiner has rejected claims 9-21, 24-33 and 36-42 under 35 U.S.C. 103(a), stating that these claims are unpatentable over Mullins in view of Fink (US Patent No. 6,490,590 - hereinafter called "Fink"). Applicant respectfully requests reconsideration of the rejection based on the reasons set out below.

As discussed above, Mullins does not disclose any metadata model having layers, especially, a data access layer, a business layer and a package layer. Thus, Mullins does not disclose or suggest any transformations for refining or constructing model objects as recited in claim 9. Especially, as the Examiner has indicated, Mullins does not teach the refining the business rules.

The Examiner has attempted to overcome this deficiency of Mullins by combining Fink with Mullins.

As discussed in the previous response, Fink discloses generation of a logical data model and physical data model. As shown in Figure 3A, only after the logical data model (LDM) is created (310), a physical data model (PDM) is created (314) "using the GDM tool 300 and the LDM resulting from step 310" (column 6, lines 41-43). In Fink's method, the physical model having a lower degree of abstraction is created after the logical model having a higher degree of abstraction is created. This is totally opposite to the transformations carried out by the metadata model transformer of the present invention, as recited in independent claims 9 and 36.

The transformer as recited in claim 9 comprises "data access to business model transformations" and "business to package model transformations". The data access to business model transformations construct business model objects (i.e., objects of a higher degree of abstraction) based on the data access model objects (i.e., objects of a lower degree of abstraction). Similarly, the business to package model transformations constructs package model objects (i.e., objects of a higher degree of abstraction) based on the business model objects (i.e., objects of a lower degree of abstraction). Fink does not disclose any transformation which can construct model objects of a higher degree of abstraction based on model objects of a lower degree of abstraction.

The Examiner has referred to the section in Fink that reads "SME refines the business rule metadata to reflect the client's business" (column 8, lines 20-22). This step is illustrated as box 326 labeled "modify metadata" in Figure 3B as a step that is carried out after the creation of the physical data model (314) and other steps. This modification of metadata is external modification carried out by a Subject Matter Expert (SME), i.e., a human. Fink does not disclose or suggest use of any transformation to refine business rules.

Accordingly, Applicant respectfully submits that claims 9 and 36 cannot be rendered obvious by Mullin and Fink.

Dependent claims 10-21, 24-33 and 37-42 recite elements or steps of transformations recited in claims 9 or 36. As neither Mullins nor Fink discloses any transformation, neither Mullins nor Fink discloses or suggests those elements or steps.

Therefore, even if one skilled in the art combines Mullins and Fink, he would still fail to provide a metadata model transformer or method for transforming a metadata model as recited in claims 9 and 36 and their dependent claims. Thus, the present invention as recited in these claims have been patentably distinguished over Mullins and Fink.

Rejection under 35 USC 103(a) to claims 22 and 23

The Examiner has further rejected claims 22 and 23 under 35 U.S.C. 103(a) stating that these claims are unpatentable over Mullins et al in view of Fink and Henninger et al (US Patent No. 5,499,371 - hereinafter called "Henninger").

Claims 22 and 23 depend on claim 21 which depends on claim 9. As discussed above, claim 9 has patentably distinguished over Mullins and Fink .

As discussed in the previous response, Henninger et al discloses an apparatus for using an object model of an object-oriented application to automatically map information between an object-oriented application and a structured database, e.g., a relational database. As described on column 8, lines 48-53, for each one-to-one and one-to-many relationship in the object model, a

PAGE 21/23 * RCVD AT 9/28/2004 4:02:29 PM [Eastern Daylight Time] * SVR:USPTO-EFXRF-1/14 * DNIS:8729306 * CSID:7709840098 * DURATION (mm-ss):06-22

foreign key column or foreign key columns are added to the database table schema in the appropriate table of the database schema, and for each many-to-many relationship in the object model, a separate join table is added to the database schema. In these cases, Henninger's method constructs a database schema and a transform, using the object model as input.

The object model of Henninger is not transformed. As shown in Figures 1 and 3, Henninger's method software 15 accepts object model 20 and accepts or creates database schema 30 and transform 50, and using these three elements as input, the method automatically generates source code (column 5, lines 63-65; column 7, lines 29-31). As seen in Step D of Figure 3, Henninger's method constructs a database schema 30 and a transform 50 derived from the object model 20 (column 7, lines 53-55). Thus, Henninger simply uses the input object model 20, and does not modify or transform the object model 20 as part of the process. This is contrary to the present invention that transforms the metadata model. Therefore, the present invention as claimed is totally different from Henninger.

Therefore, claims 22 and 23 are also patentably distinguished over Mullins, Fink and Henninger.

PAGE 22/23 * RCVD AT 9/28/2004 4:02:29 PM [Eastern Daylight Time] * SVR:USPTO-EFXRF-1/14 * DNIS:8729306 * CSID:7709840098 * DURATION (mm-ss):06-22

CONCLUSION

In conclusion, Applicant respectfully submits that the present invention as claimed in the claims is patentably distinguished over any combination of the cited references. The Applicant respectfully requests a Notice of Allowance be issued in this case. Should there be any further questions or concerns, the Examiner is urged to telephone the undersigned to expedite prosecution.

Respectfully submitted,
GARDNER GROFF, P.C.

Daniel J. Santos
Reg. No. 40,158

GARDNER GROFF, P.C.
Paper Mill Village, Building 23
600 Village Trace, Suite 300
Marietta, Georgia 30067
Phone: 770.984.2300
Fax: 770.984.0098

PAGE 23/23 * RCVD AT 9/28/2004 4:02:29 PM [Eastern Daylight Time] * SVR:USPTO-EFXRF-1/14 * DNIS:8729306 * CSID:7709840098 * DURATION (mm-ss):06-22